BITS F464 Machine Learning

Assignment 2

Date : 9th April, 2020

Team Members

Durba Satpathi	2019A7PS0972H
R Adarsh	2019A7PS0230H
Navdeep Singh	2017B5A71675H

Problem 2B Artificial Neural Networks

2.1. Model Description and implementation

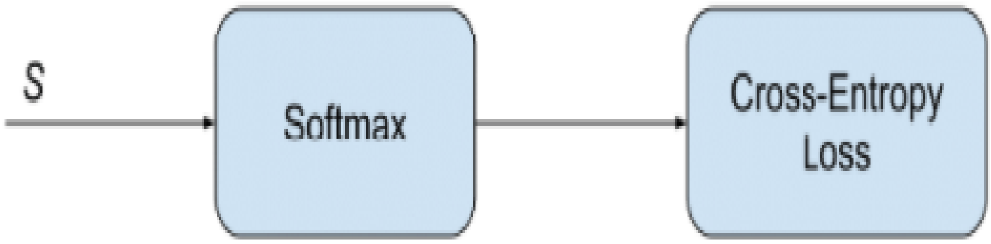
Dataset Given: The given dataset contains 2000 instances and each input instance contains 6 attributes and the target attribute which denotes the class of the instance. Each instance belongs to a class between 1,10.

Data Processing:

1) The data has been split into 70% (train set) and 30% (test set)
2) The attributes have been standardized by removing mean and scaling to unit variance. The standard score x is given by,
   x = (x-u)/s
   where u is feature mean and s is the standard deviation.
3) Since we are dealing with multiclass classification, the label vector y has been converted to a one-hot encoded vector which will be used to train the model.

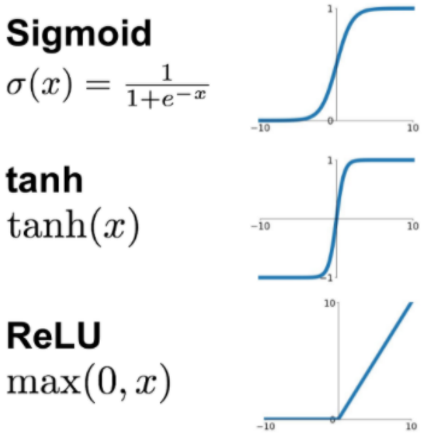
Loss Function:

The loss function used is categorical cross entropy (also known as softmax loss) . It is generally used for multi-class classification problems. If we use this loss, we train the model to output the probability over K classes.



$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \qquad CE = -\sum_i^C t_i log(f(s)_i)$$

Activation functions:
We have trained the model with different activation functions like tanh ,ReLU,sigmoid for the hidden layers and softmax activation for the output layer.



Forward Propagation:
1) First we initialize the parameters randomly.
2) The input layer provides the initial information that then propagates to hidden neurons at each layer and then finally produces the output y.
3) Feed Forward: Given the inputs from previous layers, each layer computes an affine transformation and then applies some non linear activation on it element wise.
   Z = W.T @ X + b
   A = h(Z)
   where W is weight vector
   b is the bias term
   h() is non linear activation

Back Propagation:
1) Passes the information backward from the cost through the network to compute the gradient.
2) We move backward through the layers and for each l = L-1,L-2,....2 layers the error is back propagated.
$$\delta^{x,l} = ((w^{l+1})^T \delta^{x,l+1}) \odot \sigma'(z^{x,l}).$$
3) Gradient Descent: For each l= L-1,L-2,.... 2, the parameters are updated according to,
$$w^l \rightarrow w^l - \frac{\eta}{m} \sum_x \delta^{x,l} (a^{x,l-1})^T$$
$$b^l \rightarrow b^l - \frac{\eta}{m} \sum_x \delta^{x,l}.$$
Note: In our implementation, we have done stochastic gradient descent which basically processes the training samples in mini-batches of size 1.

2.2. Description of your chosen hyper-parameters for the model such as number of hidden layers, number of units per layer, activation functions for each layer and learning rate

A. Two Layer Neural Network [input----hidden----output]:

- One Hidden Layer
- 6 units in input layer since number of attributes=6
- 32 units in hidden layer
- Experimented with tanh and ReLU activation in a hidden layer.
- 10 units in the output layer (since K=10).
- Learning rate = 0.0015

B. Three Layer Neural Network [input----hidden1----hidden2----output]:

- 2 Hidden Layers
- 6 units in input layer since number of attributes=6
- 32 units in first hidden layer
- 32 units in second hidden layer
- Experimented with tanh and ReLU activation in a hidden layer.
- 10 units in the output layer (since K=10).
- Learning rate = 0.0015

2.3 The final train and test metrics (loss and accuracy) with one hidden layer and two hidden layers:

One hidden layer architecture:

[6,32,10]
lr=0.0015,iteration=10000,activation='tanh'

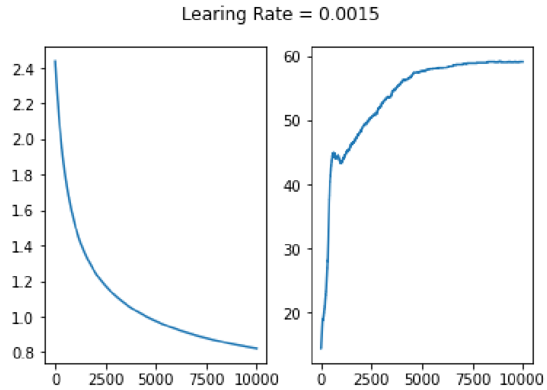
Training Accuracy : 64.86 %
Training Loss : 0.81
Testing Accuracy : 63.17 %

Two hidden layer architecture:

[6,32,32,10]
lr=0.05,iteration=1000,activation='relu'
Training accuracy**: 73.57 %**
Training Loss : 0.75
Testing Accuracy: **71.33 %**

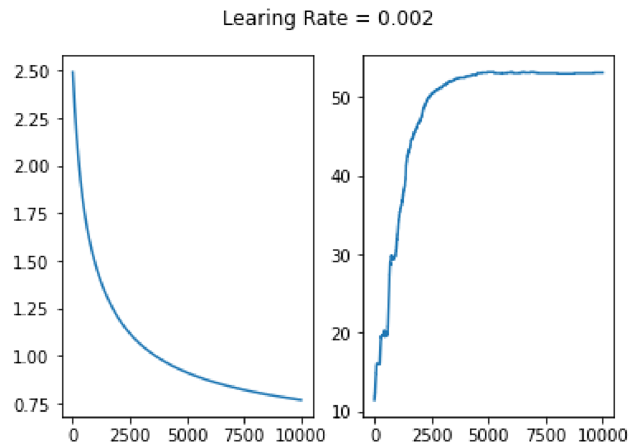
2.4 Plots of accuracy for three different learning rates for ANN with one hidden layer and two hidden layers.

1 hidden layer: 6 - 32 - 10

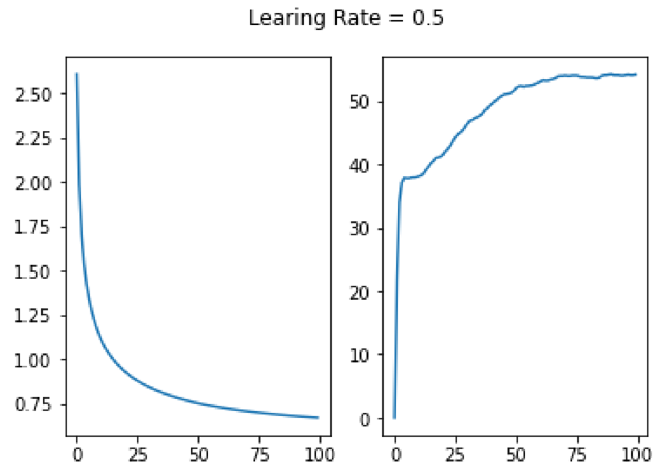


Training Accuracy : 64.86 %
Testing Accuracy 63.17 %

-

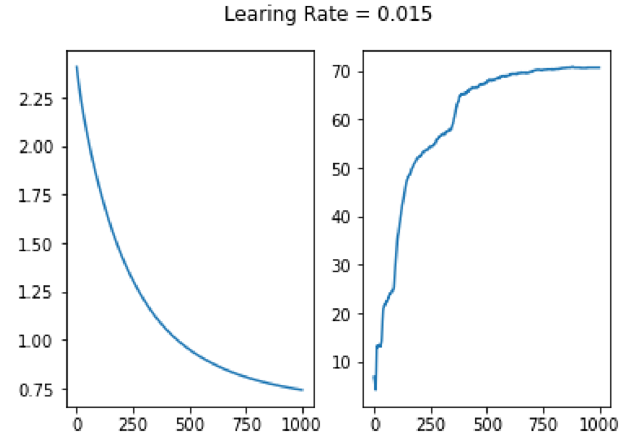


Training accuracy-56.36 %
Testing Accuracy-56.87 %

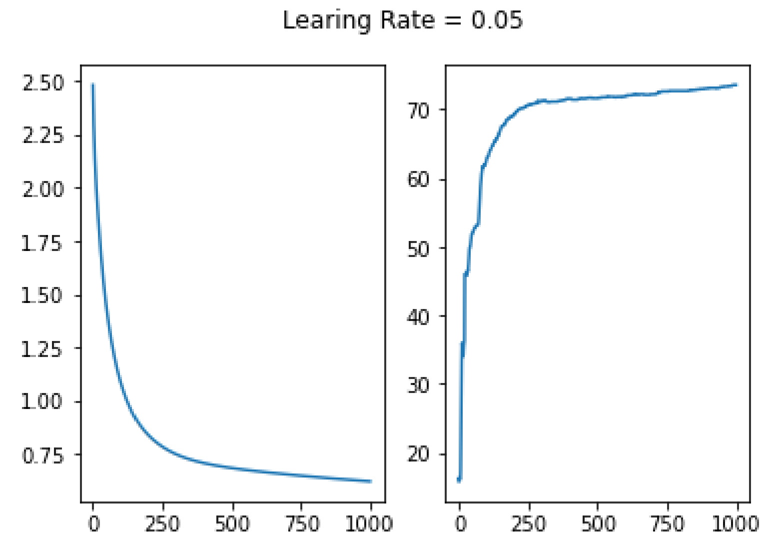


Training accuracy: 57.14 %
Testing Accuracy: 56.67 %

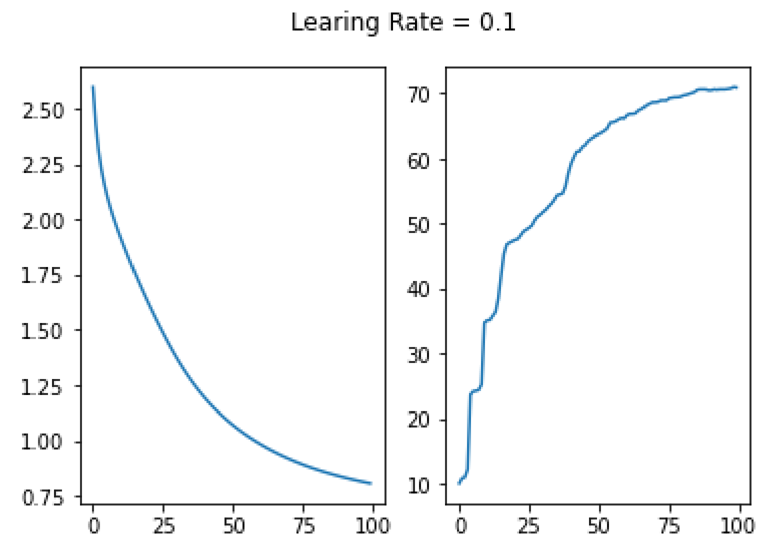
2 hidden layers: 6 - 32 - 32 -10



Training accuracy: 71.21 %
Testing Accuracy: 69.33 %



Training accuracy**: 73.57 %**
Testing Accuracy: **71.33 %**



Training accuracy**: 70.93 %**
Testing Accuracy: **68.0 %**